

ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE PSEUDOM

OBJECT MODULE PLACED IN PSEUD1.OBJ

COMPILER INVOKED BY: :F0: PSEUD1.PLM DEBUG OPTIMIZE(2) DATE(FEBRUARY 2, 1982) XREF

```

1      %title ('PSEUDO INSTRUCTION MODULE-1')
      pseudom:
      do;

/*

      modified 4/9/81  R. Silberstein
      modified 4/15/81 R. Silberstein
      modified 5/7/81  R. Silberstein
      modified 7/24/81 R. Silberstein
      modified 8/26/81 R. Silberstein
      modified 3/19/81 R. Silberstein

*/

/*
      This is the module to perform the decoding of
      all legal pseudo instructions of the assembler.
      There is one subroutine for each corresponding
      pseudoinstruction.
*/

      %include (:f1:macro.lit)
= %nolist
      %include (:f1:struc.lit)
= %nolist
      %include (:f1:equals.lit)
= %nolist
      %include (:f1:pseud1.x86)
= %nolist
      %include (:f1:outp.lit)
= %nolist
      %include (:f1:subr2.ext)
= %nolist
      %include (:f1:print.ext)
= %nolist
      %include (:f1:scan.ext)
= %nolist
      %include (:f1:symb.ext)
= %nolist
      %include (:f1:expr.ext)
= %nolist
      %include (:f1:termod.ext)
= %nolist
      %include (:f1:outp.ext)
= %nolist
      %include (:f1:global.ext)
= %nolist

```

CP/M-86 v.1.1 (Vol. IV)
 COPYRIGHT © 1981, 1982
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # C81-162

```

%object
/***** COMMON SUBROUTINES *****/

/* routine to test if rest of line is either a comment or empty -
   if not, print error message - skip rest of line */

130 1 test$emptyline: proc;
131 2     if not emptyline then call errmsg(end$of$line$err);
133 2     call skip$rest$of$line;
134 2 end test$emptyline;

/* list current address in front of printline */

135 1 listcip: proc PUBLIC;
136 2     if (prefixptr=0) and (pass < 0) then$do
138 3         call hex2out(cip,.prefix(1));
139 3         prefixptr=6;
140 3     endif;
141 2 end list$cip;

/* common routine for ORG and RS (reserve storage pseudo) */

142 1 orgrs: proc (disp,typ);
143 2     dcl disp addr,typ byte,oper operandstruc at (.operands(0));

144 2     if noforwardexpr(.oper) then$do /* evaluate operand */
146 3         if oper.type=number then$do
148 4             currentsymbol.length=oper.offset;
149 4             cip=disp+oper.offset*typ; /* compute new instruction pointer */
150 4             call test$emptyline;
151 4             return;
152 4         endif;
        endif;

/* error in expression */

154 2     call errmsg(pseudooperr);
155 2     call skip$rest$of$line;
156 2 end orgrs;

/* perform handling for CSEG,DSEG,SSEG,ESEG routines */

157 1 segmentrout: proc (p1,p2,p3,segr);
158 2     dcl segr byte,(p1,p2,p3) address,
        currentseg based p1 addr,
        segspecified based p2 byte,
        cipsave based p3 addr,
        oper operandstruc at (.operands(0)),
        low byte at (.csegvalue),high byte at (.csegvalue+1);

159 2 emit: proc;
160 3     dcl datatab(4) byte data (ESvalue,CSvalue,SSvalue,DSvalue);
161 3     call emitcodebyte(high,datatab(segr));
162 3     call emitcodebyte(low,datatab(segr));
163 3     call hex2out(csegvalue,.prefix(3)); /* print value on print line */
164 3     prefixptr=7;

```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 570

PACIFIC GROVE, CA 93950

SER. =

```

165 3      end emit;

166 2      do case csegtype;      /* save current segment attributes */

167 3          do; cureseg=csegvalue; espec=csegspec; escip=cip; end; /* ES */
172 3          do; curcseg=csegvalue; cspec=csegspec; cscip=cip; end; /* CS */
177 3          do; cursseg=csegvalue; sspec=csegspec; sscip=cip; end; /* SS */
182 3          do; curdseg=csegvalue; dspec=csegspec; dscip=cip; end; /* DS */
187 3      end$case;

188 2      if emptyline then$do /* allow no parameter */
190 3          call skip$rest$of$line;
191 3          csegvalue=0;
192 3          csegtype=sgr;
193 3          csegspec=false; /* no segment value specified */
194 3          cip=0;
195 3          return;
196 3      end$if;

197 2      if specialtoken('$') then$do /* allow '$' */
199 3          csegtype=sgr; /* pick up previous values */
200 3          csegspec=segspecified;
201 3          csegvalue=currentseg;
202 3          cip=cipsave;
203 3          if csegspec then call emit;
205 3          call scon; /* skip $ */
206 3          call test$emptyline;
207 3          return;
208 3      end$if;

209 2      if expression(.oper) then$do /* operand must be expression */
211 3          if oper.stype=number then$do
213 4              csegvalue=oper.offset; /* pick up segment value */
214 4              csegtype=sgr;
215 4              csegspec=true; /* value is specified */
216 4              cip=0;
217 4              call emit;
218 4              call test$emptyline;
219 4              return;
220 4          end$if;
221 3          /* must be illegal operand */
222 2          call skip$rest$of$line;
223 2          call errmsg(pseudooperr);

224 2      end segmentout;

/* common routine for DB,DW and DD */

225 1      DB$DW$DD$common: proc(n);
226 2          dcl(n,continue) byte,lq addr;
227 2          DECLARE EP BYTE;

228 2      item: proc(n); /* find one element of element list */
229 3          dcl (n,i,errorprinted) byte,

```

COPYRIGHT © 1981, 1982
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

        oper operandstruc at (.operands(0)),
        low   byte at (.oper.offset),
        high  byte at (.oper.offset+1),
        seglow byte at (.oper.segment),
        seghigh byte at (.oper.segment+1);
230 3      emit: proc (outputbyte);
231 4          dcl outputbyte byte,
                datatab(4) byte data (ESdata,CSdata,SSdata,DSdata);
232 4          call emitcodebyte(outputbyte,datatab(csegtype));
233 4      end emit;

234 3      locexpr: proc byte;
235 4          if expression(.oper) then$do
237 5              i=oper.stype;
238 5              if (i=number) or (i=variable) or (i=lab) then return true;
                end$if;
241 4          return false;
242 4      end locexpr;

243 3      DBhandle: proc;
244 4          if (token.type=string) and (acclen > 1) then$do
246 5              lg=lg+acclen-1;
247 5              i=0ffh;
248 5              do while (i:=i+1) < acclen;
249 6                  call emit(accum(i));
250 6              end$while;
251 5              oper.stype=number; /* dummy */
252 5              call scan; /* skip string */
253 5          else$do
255 5              if locexpr then$do
257 6                  call emit(low);
258 6              else$do
260 6                  call emit(0);
261 6                  call errmsg(illexprelem);
262 6              end$if;
263 5          end$if;
264 4      end DBhandle;

265 3      DWhandle: proc;
266 4          if locexpr then$do
268 5              call emit(low);
269 5              call emit(high);
270 5          else$do
272 5              call emit(0);
273 5              call emit(0);
274 5              call errmsg(illexprelem);
275 5          end$if;
276 4      end DWhandle;

277 3      DDhandle: proc;
278 4          if locexpr then$do
280 5              if oper.stype <> number then$do
282 6                  if (oper.sflag and segmbit) <> 0 then$do
284 7                      call emit(low);
285 7                      call emit(high);
286 7                      call emit(seglow);
287 7                      call emit(seghigh);

```

COPYRIGHT © 1981, 1982
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

238 7      return;
239 7      else$do
291 7          call errmsg(misseqminfo);
292 7      end$if;
293 6      end$if;
          end$if;
295 4      do i=0 to 3; call emit(0); end$do; /* dummy */
298 4      call errmsg(illexprelem);
299 4      end DDhandle;

          /* ITEM main program */
300 3      lg=lg+1;
301 3      do case n;
302 4          call DBhandle;
303 4          call DWhandle;
304 4          call DDhandle;
305 4      end$case;
306 3      if specialtoken(',') then$do
308 4          call scan;
309 4          continue=true;
310 4      else$do
312 4          if emptyline then$do
314 5              call skip$rest$of$line;
315 5          else$do
317 5              CALL ERRMSG (ENDOFLINEERR);
318 5              CALL SKIPRESTOFLINE;
319 5          end$if;
320 4      end$if;
321 3      end item;

          /* DB$DW$DD$common main program */
322 2      CALL LISTCIP;
323 2      EP = FALSE;
324 2      lg=0;
325 2      continue=true;
326 2      do while continue;
327 3          errorprinted=false;
328 3          continue=false;
329 3          call item(n);
330 3          EP = EP OR ERRORPRINTED;
331 3      end$while;
332 2      currentsymbol.length=lg;
333 2      ERRORPRINTED = EP; /* SO SOURCE LINE IS ECHOED IF ERROR */
334 2      end DB$DW$DD$common;

```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

#eject

/***** PSEUDO SUBROUTINES *****/

```

335 1  DBrout: proc public;
336 2      call DB$DW$DD$common(0);
337 2  end DBrout;

338 1  DWrout: proc public;
339 2      call DB$DW$DD$common(1);
340 2  end DWrout;

341 1  DDrout: proc public;
342 2      call DB$DW$DD$common(2);
343 2  end DDrout;

344 1  RSrout: proc (typ) public;
345 2      dcl typ byte;
346 2      call listcip; /* list current address on printline */
347 2      call orgrs(cip,typ); /* cip = cip + typ * expression */
348 2  end RSrout;

349 1  CSEGrout: proc public;
350 2      call segmentrout(.curcseg,.cspec,.cscip,rcs);
351 2  end CSEGrout;

352 1  DSEGrout: proc public;
353 2      call segmentrout(.curdseg,.dspec,.dscip,rds);
354 2  end DSEGrout;

355 1  SSEGrout: proc public;
356 2      call segmentrout(.cursseg,.sspec,.sscip,rss);
357 2  end SSEGrout;

358 1  ESEGrout: proc public;
359 2      call segmentrout(.cureseg,.espec,.escip,res);
360 2  end ESEGrout;

361 1  ORGrout: proc public;
362 2      call orgrs(0,byt); /* cip = 0 + expression */
363 2  end ORGrout;

364 1  EQUrout: proc public;
365 2      dcl oper operandstruc at (.operands(0)),
          macdefpt based codemacroptr address;
366 2      codempossible: proc byte;
367 3          return (nextch=cr or nextch=' ');
368 3      end codempossible;

369 2      do case pass;

370 3          do; /* pass 0 */
371 4          if codempossible and
              findcodemacro(acclen,.accum(0),.codemacroptr) then do
373 5              currentsymbol.stype=code;
374 5              call enterattributes(symtabadr,.currentsymbol);
375 5              if not newmacro(acclensave,.accumsave,macdefpt) then

```

COPYRIGHT © 1981, 1982
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

376 5      fullsymtab=true;
      else$do
379 5          nooper=0; /* find normal operand expression */
380 5          IF NOFORWARDOPER THEN$DO
382 6              call enterattributes(symtabadr,.operands(0));
383 6              call skip$rest$of$line;
384 6          else$do
386 6              currentsymbol.stype=undefsym;
387 6              call enterattributes(symtabadr,.currentsymbol);
388 6              call skip$rest$of$line;
389 6          end$if;
390 5      end$if;
391 4      end;

392 3      do; /* pass 1 */
393 4      if currentsymbol.stype <> code then$do /* update symbol value */
395 5          nooper=0;
396 5          IF NOFORWARDOPER THEN$DO
398 6              call enterattributes(symtabadr,.operands(0));
399 6          end$if;
401 4      end$if;
402 4      call skip$rest$of$line;
403 3      end;

403 3      do; /* pass 2 - scan to produce possible error messages */
404 4      if currentsymbol.stype=code then$do
406 5          call scan;
407 5      else$do
409 5          nooper=0;
410 5          IF NOT NOFORWARDOPER OR (CURRENTSYMBOL.STYPE = ERROR) THEN$DO
412 6              call errmsg(pseudooperr);
413 6              call skip$rest$of$line; /* only one error message */
414 6          else$do
416 6              prefixptr=7;
417 6              call hex2out(oper.offset,.prefix(3));
418 6          end$if;
419 5      end$if;
420 4      call test$emptyline;
421 4      end;

422 3      end$case;
423 2      end EQUROUT;

424 1      end$module pseudom;

```

COPYRIGHT © 1981, 1982
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

 DEFN ADDR SIZE NAME, ATTRIBUTES, AND REFERENCES

127	0000H	4	ABSADDR.	BYTE ARRAY(4) EXTERNAL(77)	
127	0000H	1	ACCLEN.	BYTE EXTERNAL(81)	244 246 248 371
127	0000H	1	ACCLNSAVE.	BYTE EXTERNAL(84)	375
127	0000H	80	ACCUM.	BYTE ARRAY(80) EXTERNAL(82)	249 371
127	0000H	80	ACCUMSAVE.	BYTE ARRAY(80) EXTERNAL(83)	375
2			ADDR.	LITERALLY	12 18 36 39 42 45 77 78 84 87 90
				93 94 99 102 105 127 143 158 226 229 365	
26	0000H		ALPHANUMERIC.	PROCEDURE BYTE EXTERNAL(5) STACK=0000H	
29	0000H		ASCIICHAR.	PROCEDURE BYTE EXTERNAL(6) STACK=0000H	
104	0000H	2	ASCIIPTR.	WORD PARAMETER	105
229	0008H	1	BASEINDEX.	BYTE MEMBER(OPER)	
365	0008H	1	BASEINDEX.	BYTE MEMBER(OPER)	
127	0008H	1	BASEINDEX.	BYTE MEMBER(CURRENTSYMBOL)	
138	0008H	1	BASEINDEX.	BYTE MEMBER(OPER)	
127	0008H	1	BASEINDEX.	BYTE MEMBER(OPERANDS)	
143	0008H	1	BASEINDEX.	BYTE MEMBER(OPER)	
3			BASEREBIT.	LITERALLY	
3			BASEREGCOUNT.	LITERALLY	
3			BREGBIT.	LITERALLY	
3			BREGCOUNT.	LITERALLY	
5			BYT.	LITERALLY	362
20	0000H	1	CH.	BYTE PARAMETER	21
29	0000H	1	CH.	BYTE PARAMETER	30
124	0000H	1	CH.	BYTE PARAMETER	125
47	0000H	1	CH.	BYTE PARAMETER	48
26	0000H	1	CH.	BYTE PARAMETER	27
23	0000H	1	CH.	BYTE PARAMETER	24
17	0000H	1	CH.	BYTE PARAMETER	18
32	0000H	1	CH.	BYTE PARAMETER	33
127	0000H	2	CIP.	WORD EXTERNAL(46)	138 149 170 175 180 185 194 202 216
				347	
158	0000H	2	CIPSAVE.	WORD BASED(P3)	202
4			CODE.	LITERALLY	373 393 404
127	0000H	1	CODEMACROFLAG.	BYTE EXTERNAL(103)	
3			CODEMACROHEAD.	LITERALLY	
127	0000H	2	CODEMACROPTR.	WORD EXTERNAL(98)	365 371 375
366	0511H	29	CODEMPOSSIBLE.	PROCEDURE BYTE STACK=0004H	371
226	0000H	1	CONTINUE.	BYTE	309 325 326 328
2			CR.	LITERALLY	367
127	0000H	2	COCIP.	WORD EXTERNAL(51)	175 350
10			CSDATA.	LITERALLY	231
349	03BCH	23	CSEGROUT.	PROCEDURE PUBLIC STACK=0010H	
127	0000H	1	CSEGSPEC.	BYTE EXTERNAL(49)	169 174 179 184 193 200 203 215
127	0000H	1	CSEGTYP.	BYTE EXTERNAL(47)	166 192 199 214 232
127	0000H	2	CSEGV.	WORD EXTERNAL(48)	158 163 168 173 178 183 191 201 213
127	0000H	1	CSPEC.	BYTE EXTERNAL(58)	174 350
10			CSVALUE.	LITERALLY	160
127	0000H	2	CURCSEG.	WORD EXTERNAL(54)	173 350
127	0000H	2	CURDSEG.	WORD EXTERNAL(55)	183 353

COPYRIGHT © 1981, 1982
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. #

127	0000H	2	CURESEG.	WORD EXTERNAL(57)	168	359																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		</
-----	-------	---	------------------	-------------------	-----	-----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	----

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. 17

78	0000H	2	ENDOSYMBTAB . . .	WORD EXTERNAL(26)	
2			ENDPROC.	LITERALLY	
2			ENDWHILE	LITERALLY	
95	0000H		ENTERATTRIBUTES. .	PROCEDURE EXTERNAL(33) STACK=0000H	374 382 387 398
127	0000H	1	EOFSET	BYTE EXTERNAL(85)	
10			EOFTYPE.	LITERALLY	
227	000BH	1	EP	BYTE 323 330 333	
35	0000H		EQUAL.	PROCEDURE BYTE EXTERNAL(8) STACK=0000H	
364	0427H	234	EQUROUT.	PROCEDURE PUBLIC STACK=000CH	
117	0000H		ERRMSG	PROCEDURE EXTERNAL(41) STACK=0000H	132 154 223 261 274 291
				298 317 412	
117	0000H	1	ERRNO.	BYTE PARAMETER 118	
4			ERROR.	LITERALLY 410	
127	0000H	1	ERRORPRINTED . . .	BYTE EXTERNAL(88) 327 330 333	
229	000DH	1	ERRORPRINTED . . .	BYTE	
127	0000H	2	ERRORS	WORD EXTERNAL(78)	
127	0000H	2	ESCIP.	WORD EXTERNAL(50) 170 359	
10			ESDATA	LITERALLY 231	
358	0401H	23	ESEGROUT	PROCEDURE PUBLIC STACK=0010H	
127	0000H	1	ESPEC.	BYTE EXTERNAL(61) 169 359	
10			ESVALUE.	LITERALLY 160	
111	0000H		EXPRESSION	PROCEDURE BYTE EXTERNAL(39) STACK=0000H	209 235
2			FALSE.	LITERALLY 193 241 323 327 328	
14	0000H	2	FCBADR	WORD PARAMETER 15	
14	0000H		FILEABORT.	PROCEDURE EXTERNAL(1) STACK=0000H	
3			FILEISTRUCTURE . .	LITERALLY	
3			FILEDSTRUCTURE . .	LITERALLY	
17	0000H		FILL	PROCEDURE EXTERNAL(2) STACK=0000H	
98	0000H		FINDCODEHACKO. . .	PROCEDURE BYTE EXTERNAL(34) STACK=0000H	371
89	0000H		FINDSYMBOL	PROCEDURE BYTE EXTERNAL(31) STACK=0000H	
127	0000H	2	FIRSTMACROPTR. . .	WORD EXTERNAL(89)	
2			FOREVER.	LITERALLY	
2			FORMFEED	LITERALLY	
77	0000H	2	FREETPT	WORD EXTERNAL(25)	
127	0000H	1	FULLSYMBTAB. . . .	BYTE EXTERNAL(91) 376	
92	0000H		GETATTRIBUTES. . .	PROCEDURE EXTERNAL(32) STACK=0000H	
128	0000H		GLOBALINIT	PROCEDURE EXTERNAL(104) STACK=0000H	
127	0000H	5	HELP	BYTE ARRAY(5) EXTERNAL(79)	
38	0000H		HEX1OUT.	PROCEDURE EXTERNAL(9) STACK=0000H	
41	0000H		HEX2OUT.	PROCEDURE EXTERNAL(10) STACK=0000H	138 163 417
158	0001H	1	HIGH	BYTE EXTERNAL(48) AT 161	
229	0007H	1	HIGH	BYTE EXTERNAL(97) AT 269 285	
127	0000H	1	I.	BYTE EXTERNAL(100)	
229	000CH	1	I.	BYTE 237 238 247 248 249 295	
4			IDENT.	LITERALLY	
127	0000H	1	IFLEVEL.	BYTE EXTERNAL(93)	
127	0000H	1	IFLIST	BYTE EXTERNAL(64)	
9			ILLEXPRELEN. . . .	LITERALLY 261 274 298	
127	0000H	1	INCLUDEDEFAULT . .	BYTE EXTERNAL(102)	
127	0000H	1	INCLUDEON.	BYTE EXTERNAL(92)	
3			INDEXREGBIT. . . .	LITERALLY	
3			INDEXREGCOUNT. . .	LITERALLY	
2			INIT	LITERALLY	
10			INTELDATA.	LITERALLY	
127	0000H	1	INTELHEXON	BYTE EXTERNAL(86)	
10			INTELSEGMENT . . .	LITERALLY	
3			IREGBIT.	LITERALLY	

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P.O. BOX 879

PACIFIC GROVE, CA 93950

SER. #

[illegible]

229	0006H	2	OFFSET	WORD MEMBER(OPER)	229					
8			OGE	LITERALLY						
8			OGT	LITERALLY						
8			OLAST	LITERALLY						
8			OLE	LITERALLY						
8			OLENGTH	LITERALLY						
8			OLT	LITERALLY						
8			ONOD	LITERALLY						
8			ONE	LITERALLY						
8			ONOT	LITERALLY						
8			DOFFSET	LITERALLY						
8			DOOR	LITERALLY						
365	0000H	9	OPER	STRUCTURE EXTERNAL(97) AT	417					
229	0000H	9	OPER	STRUCTURE EXTERNAL(97) AT	229	235	237	251	280	282
158	0000H	9	OPER	STRUCTURE EXTERNAL(97) AT	209	211	213			
143	0000H	9	OPER	STRUCTURE EXTERNAL(97) AT	144	146	148	149		
107	0000H		OPERAND	PROCEDURE BYTE EXTERNAL(37) STACK=0000H						
127	0000H	36	OPERANDS	STRUCTURE ARRAY(4) EXTERNAL(97)			143	158	229	365 382 398
3			OPERANDSTRUC	LITERALLY	127	143	158	229	365	
4			OPERATOR	LITERALLY						
8			OPTR	LITERALLY						
361	0413H	15	ORGROUT	PROCEDURE PUBLIC STACK=0012H						
142	003AH	60	ORGRS	PROCEDURE STACK=000EH	347	362				
8			OSEG	LITERALLY						
8			OSHL	LITERALLY						
8			OSHORT	LITERALLY						
8			OSHR	LITERALLY						
8			OTYPE	LITERALLY						
230	0004H	1	OUTPUTBYTE	BYTE PARAMETER AUTOMATIC	231	232				
11	0000H		OUTTEXT	PROCEDURE EXTERNAL(0) STACK=0000H						
8			OXOR	LITERALLY						
157	0006H	2	P1	WORD PARAMETER	158	201				
157	0004H	2	P2	WORD PARAMETER	158	200				
157	0002H	2	P3	WORD PARAMETER	158	202				
127	0000H	1	PAGESIZE	BYTE EXTERNAL(71)						
127	0000H	1	PASS	BYTE EXTERNAL(45)	136	369				
7			PCODEMACRO	LITERALLY						
7			PCSEG	LITERALLY						
7			PDB	LITERALLY						
7			PDBIT	LITERALLY						
7			PDD	LITERALLY						
7			PDSEG	LITERALLY						
7			PDW	LITERALLY						
7			PEJECT	LITERALLY						
7			PEND	LITERALLY						
7			PENDIF	LITERALLY						
7			PENDM	LITERALLY						
7			PEQU	LITERALLY						
7			PESEG	LITERALLY						
7			PIF	LITERALLY						
7			PIFLIST	LITERALLY						
7			PINCLUDE	LITERALLY						
7			PLIST	LITERALLY						
7			PMODRM	LITERALLY						
7			PNOIFLIST	LITERALLY						
7			PNOLIST	LITERALLY						
7			PNOSEGFIX	LITERALLY						

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____

COPYRIGHT © 1931, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. #

104	0000H	2	RETURNPTR.	WORD PARAMETER	105				
8			RIGHTBRACKET . . .	LITERALLY					
6			RSI.	LITERALLY					
6			RSP.	LITERALLY					
344	03A3H	20	RSROUT	PROCEDURE PUBLIC STACK=0014H					
6			RSS.	LITERALLY	356				
35	0000H	2	S.	WORD PARAMETER	36				
127	0000H	12	SAVESOURCE	BYTE ARRAY(12) EXTERNAL(67)					
65	0000H		SCAN	PROCEDURE EXTERNAL(20) STACK=0000H		205	252	308	406
63	0000H		SCANINIT	PROCEDURE EXTERNAL(19) STACK=0000H					
229	0005H	1	SEGHIGH.	BYTE EXTERNAL(97) AT	287				
229	0004H	1	SEGLOW	BYTE EXTERNAL(97) AT	286				
3			SEGBIT.	LITERALLY	282				
3			SEGMCOUNT.	LITERALLY					
365	0004H	2	SEGMENT.	WORD MEMBER(OPER)					
229	0004H	2	SEGMENT.	WORD MEMBER(OPER)	229				
158	0004H	2	SEGMENT.	WORD MEMBER(OPER)					
143	0004H	2	SEGMENT.	WORD MEMBER(OPER)					
127	0004H	2	SEGMENT.	WORD MEMBER(OPERANDS)					
127	0004H	2	SEGMENT.	WORD MEMBER(CURRENTSYMBOL)					
157	0076H	274	SEGMENTROUT. . . .	PROCEDURE STACK=000CH		350	353	356	359
157	0000H	1	SEGR	BYTE PARAMETER	158	161	162	192	199
158	0000H	1	SEGSPECIFIED . . .	BYTE BASED(P2)	200				
3			SEGTYPEBIT	LITERALLY					
3			SEGTYPECOUNT . . .	LITERALLY					
365	0003H	1	SFLAG.	BYTE MEMBER(OPER)					
229	0003H	1	SFLAG.	BYTE MEMBER(OPER)	232				
158	0003H	1	SFLAG.	BYTE MEMBER(OPER)					
143	0003H	1	SFLAG.	BYTE MEMBER(OPER)					
127	0003H	1	SFLAG.	BYTE MEMBER(OPERANDS)					
127	0003H	1	SFLAG.	BYTE MEMBER(CURRENTSYMBOL)					
127	0000H	1	SINFORN.	BYTE EXTERNAL(72)					
67	0000H		SKIPRESTOFLINE . .	PROCEDURE EXTERNAL(21) STACK=0000H		133	155	190	222
						314	318		
						383	388	401	413
72	0000H		SKIPUNTIL.	PROCEDURE BYTE EXTERNAL(23) STACK=0000H					
95	0000H	2	SOURCE	WORD PARAMETER	96				
127	0000H	80	SOURCEBUF.	BYTE ARRAY(80) EXTERNAL(73)					
127	0000H	12	SOURCENAME	BYTE ARRAY(12) EXTERNAL(66)					
127	0000H	1	SOURCEPTR.	BYTE EXTERNAL(74)					
2			SPACE.	LITERALLY					
4			SPEC	LITERALLY					
69	0000H		SPECIALTOKEN . . .	PROCEDURE BYTE EXTERNAL(22) STACK=0000H		197	306		
127	0000H	2	SSCIP.	WORD EXTERNAL(52)	180	356			
10			SSDATA	LITERALLY	231				
355	03EAH	23	SSEGROUT	PROCEDURE PUBLIC STACK=0010H					
127	0000H	1	SSPEC.	BYTE EXTERNAL(60)	179	356			
10			SSVALUE.	LITERALLY	160				
10			STARTTYPE.	LITERALLY					
98	0000H	2	STRADR	WORD PARAMETER	99				
89	0000H	2	STRADR	WORD PARAMETER	90				
86	0000H	2	STRADR	WORD PARAMETER	87				
83	0000H	2	STRADR	WORD PARAMETER	84				
4			STRING	LITERALLY	244				
2			STRUC.	LITERALLY	127	143	158	229	365
365	0002H	1	STYPE.	BYTE MEMBER(OPER)					
229	0002H	1	STYPE.	BYTE MEMBER(OPER)	237	251	280		
158	0002H	1	STYPE.	BYTE MEMBER(OPER)	211				

143	0002H	1	STYPE.	BYTE MEMBER(OPER)	146					
127	0002H	1	STYPE.	BYTE MEMBER(OPERANDS)						
127	0002H	1	STYPE.	BYTE MEMBER(CURRENTSYMBOL)		373	386	393	404	410
95	0000H	2	SYMBADR.	WORD PARAMETER	96					
92	0000H	2	SYMBADR.	WORD PARAMETER	93					
81	0000H		SYMBINIT.	PROCEDURE EXTERNAL(28) STACK=0000H						
4			SYMBOL.	LITERALLY						
127	0000H	1	SYMBOLDEVICE. . . .	BYTE EXTERNAL(69)						
3			SYMBOLHEAD.	LITERALLY						
3			SYMBOLSTRUC.	LITERALLY	127 143 158 229 365					
127	0000H	2	SYMBTABADR.	WORD EXTERNAL(95)	374 382 387 398					
79	0000H		SYMBTERMINATE. . . .	PROCEDURE EXTERNAL(27) STACK=0000H						
11	0000H	2	T.	WORD PARAMETER	12					
2			TAB.	LITERALLY						
130	0000H	23	TESTEMPTYLINE. . . .	PROCEDURE STACK=0006H	150 206 218 420					
14	0000H	2	TEXTADR.	WORD PARAMETER	15					
2			THENDO.	LITERALLY	136 144 146 188 197 209 211 235 244 255 266					
					278 280 282 306 312 371 380 393 396 404 410					
127	0000H	30	TITLE.	BYTE ARRAY(30) EXTERNAL(70)						
72	0000H	1	TOK.	BYTE PARAMETER	73					
69	0000H	1	TOK.	BYTE PARAMETER	70					
127	0000H	4	TOKEN.	STRUCTURE EXTERNAL(79)	244					
2			TRUE.	LITERALLY	215 239 309 325 376					
344	0004H	1	TYP.	BYTE PARAMETER AUTOMATIC	345 347					
142	0004H	1	TYP.	BYTE PARAMETER AUTOMATIC	143 149					
124	0000H	1	TYP.	BYTE PARAMETER	125					
127	0000H	1	TYPE.	BYTE MEMBER(TOKEN)	244					
3			TYPEBIT.	LITERALLY						
3			TYPECOUNT.	LITERALLY						
4			UDEFSYMB.	LITERALLY	386					
32	0000H		UPPER.	PROCEDURE BYTE EXTERNAL(7) STACK=0000H						
58	0000H	1	USEFACT.	BYTE PARAMETER	59					
127	0002H	2	VALUE.	WORD MEMBER(TOKEN)						
4			VARIABLE.	LITERALLY	238					
5			WRD.	LITERALLY						

MODULE INFORMATION:

```

CODE AREA SIZE      = 052EH 1326D
CONSTANT AREA SIZE  = 0008H   8D
VARIABLE AREA SIZE  = 000EH  14D
MAXIMUM STACK SIZE  = 001EH  30D
963 LINES READ
0 PROGRAM ERROR(S)

```

END OF PL/M-86 COMPILATION

